## WHAT IS CLAIMED IS:

1. A method for providing communication access between a first process and a second process comprising the steps, executed in a data processing system, of:

appending security context information for the first process in a process table;

opening a socket between the first process and the second process; and

transmitting a packet from the first process to the second process through the open socket including the security context information for the first process in the process table.

2. The method of claim 1, further comprising modifying a socket structure so as to accept the security context information.

3. The method of claim 1, further comprising:

receiving the packet at the second process through the socket;

verifying the security context information received in the packet; and

permitting use of the packet if the security context information is verified.

4. The method of claim 3, wherein verifying the security context information includes:

determining if the first and second process belong to a channel; and

accepting the transmitted packet when the first and second process belong to the channel.

5. The method of claim 4, wherein determining if the first and second process belong to a channel includes:

comparing the security context information in the received packet and security context information in another process table.

6. The method of claim 5, wherein the process table and the another process table are located on a single node.

7. The method of claim 3, wherein verifying the security context information includes:

determining whether the first and second process belong to two different linked channels; and

permitting use of the packet when the different channels are linked.

8. The method of claim 7, wherein determining whether the first and second process belong to two different linked channels includes:

initiating a process that spawns two child processes that are connected by a shared-memory region in a memory.

9. The method of claim 7, wherein permitting use of the packet includes:

decrypting the packet on a node; and

authenticating a sender associated with the first process on the node.

10. The method of claim 1, wherein appending security context information includes:

obtaining the security context information from a third process including a virtual address and a node identification; and

limiting each of the first, second and third processes to communicate with another process provided that the communicating processes share the same node identification.

11. The method of claim 1, further comprising:

modifying a network stack such that the network stack requires the security context information to be present in the socket structure to transmit.

12. A method for placing processes executed in a node in a security context, comprising the steps of:

sending a request from the node to a server to verify a username and a node identification associated with a process;

in response to the request, receiving security context information at the node from the server including a virtual address for the node;

initiating the process; and

appending the security context information and the node identification associated with the process in a process table.

13. The method of claim 12, wherein receiving security context information further includes:

receiving a key that corresponds to the node identification from the server.

14. The method of claim 13, further comprising:

encrypting a packet transmitted by the process using the key;

encapsulating the encrypted packet with a header that includes the node identification.

15. The method of claim 12, further comprising:

sending a second request from the node to the server to verify a username and node identification;

receiving additional security context information from the server, wherein the additional security context information includes a second virtual address for the node;

creating a second process; and

appending the security context information for the second process in the process table that is associated with the second process.

16. A method for providing secure communications between a first process and a second process comprising the steps, executed in a data processing system, of:

obtaining a node identification and a virtual address;

including the node identification and the virtual address in a field corresponding to the first process in a process table;

transmitting a datagram that contains the node identification and the virtual address from the first process to a socket; and

receiving the datagram at the second process that contains the node identification and a second virtual address.

17. The method of claim 16, wherein obtaining a node identification and a virtual address further includes:

modifying a socket structure in the socket so that the socket structure accepts the node identification and the virtual address; and

modifying a process table so that the table includes a node identification field and a virtual address field.

18. A system for providing communication access between a first process and a second process, comprising:

means for appending security context information for the first process in a process table;

means for opening a socket between the first process and the second process; and

means for transmitting a packet from the first process to the second process through the open socket including the security context information for the first process in the process table.

19. The system of claim 18, further comprising means for modifying a socket structure so as to accept the security context information.

20. The system of claim 18, further comprising:

means for receiving the packet at the second process through the socket;

means for verifying the security context information received in the packet; and

means for permitting use of the packet if the security context information is verified.

21. The system of claim 20, wherein means for verifying the security context information includes:

means for determining if the first and second process belong to a channel; and

means for accepting the transmitted packet when the first and second process belong to the channel.

22. The system of claim 21, wherein means for determining if the first and second process belong to a channel includes:

means for comparing the security context information in the received packet and security context information in another process table.

23. The system of claim 22, wherein the process table and the another process table are located on a single node.

24. The system of claim 20, wherein means for verifying the security context information includes:

means for determining whether the first and second process belong to two different linked channels; and

means for permitting use of the packet when the different channels are linked.

25. The system of claim 24, wherein means for determining whether the first and second process belong to two different linked channels includes:

means for initiating a process that spawns two child processes that are connected by a shared-memory region in a memory.

26. The system of claim 24, wherein means for permitting use of the packet includes:

means for decrypting the packet on a node; and

means for authenticating a sender associated with the first process on the node.

27. The system of claim 18, wherein means for appending security context information includes:

means for obtaining the security context information from a third process including a virtual address and a node identification; and

means for limiting each of the first, second and third processes to communicate with another process provided that the communicating processes share the same node identification.

28. The system of claim 18, further comprising:

means for modifying a network stack such that the network stack requires the security context information to be present in the socket structure to transmit.

29. A system for placing a process executed in a node in a security context, comprising:

a server; and

a sending node comprising:

a transmission module that transmits a request to the server to verify a user name and a node identification, and receives security context information from the server in response to the request, wherein the security context information includes a virtual address for the sender node;

memory containing a process and an associated process table; and

an appending module that appends the received security context information and the node identification for the process in the process table.

30. The system of claim 29, wherein the transmission module further receives a key that corresponds to the node identification from the server.

31. The system of claim 30, further comprising:

an encryption module that encrypts a packet transmitted by the process using the key;

an encapsulating module that encapsulates the encrypted packet with a header that includes the node identification.

32. The system of claim 29, further comprising:

a gateway that provides communication between the process and a second process executing in the node, and

wherein the transmission module further sends a second request to the server to verify a username and node identification, and receives additional security context information from the server, wherein the additional security context information includes a second virtual address for the node;

appending the security context information for the second process in a process table that is associated with the second process.

33. A system for providing secure communications between a first process and a second process, comprising:

means for obtaining a node identification and a virtual address;

means for including the node identification and the virtual address in a field corresponding to the first process in a process table;

means for transmitting a datagram that contains the node identification and virtual address from the first process to a socket; and

means for receiving the datagram at the second process that contains the node identification and a second virtual address.

34. The system of claim 33, wherein means for obtaining a node identification and a virtual address further comprises:

means for modifying a socket structure in the socket so that the socket structure accepts the node identification and the virtual address; and

means for modifying a process table so that the table includes a node identification field and a virtual address field.

35. A computer readable medium for controlling a data processing system to perform a method for providing communication access between a first process and a second process, comprising:

an appending module for appending security context information for the first process in a process table;

an opening module for opening a socket between the first process and the second process; and

a transmitting module for transmitting a packet from the first process to the second process through the open socket including the security context information for the first process in the process table.

36. The computer readable medium of claim 35, further comprising a modifying module for modifying a socket structure so as to accept the security context information.

37. The computer readable medium of claim 35, further comprising:

a receiving module for receiving the packet at the second process through the socket;

a verifying module for verifying the security context information received in the packet; and

a permitting module for permitting use of the packet if the security context information is verified.

38. The computer readable medium of claim 36, wherein the verifying module includes:

a determining module for determining if the first and second process belong to a channel; and

an accepting module for accepting the transmitted packet when the first and second process belong to the channel.

39. The computer readable medium of claim 38, wherein the determining module includes:

a comparing module that compares the security context information in the received packet and security context information in another process table.

40. The computer readable medium of claim 39, wherein the process table and the another process table are located on a single node.

41. The computer readable medium of claim 37, wherein the verifying module includes:

a determining module for determining whether the first and second process belong to two different linked channels; and

a permitting module for permitting use of the packet when the different channels are linked.

42. The computer readable medium of claim 41, wherein the determining module includes a initiating module that initiates a process that spawns two child processes that are connected by a shared-memory region in a memory.

43. The computer readable medium of claim 41, wherein the permitting module includes:

a decrypting module for decrypting the packet on a node; and

an authenticating module for authenticating a sender associated with the first process on the node.

44. The computer readable medium of claim 35, wherein the appending module includes:

an obtaining module for obtaining the security context information from a third process including a virtual address and a node identification; and

a limiting module for limiting each of the first, second and third processes to communicate with another process provided that the communicating processes share the same node identification.

45. The computer readable medium of claim 35, further comprising:

a modifying module for modifying a network stack such that the network stack requires the security context information to be present in the socket structure to transmit